AUTOMATED ENTITLEMENT VERIFICATION FOR DELIVERY OF LICENSED SOFTWARE

TABLE OF CONTENTS

5 (Provided for Examination Reference Purposes)

INVENTOR: Victor R. HERRERO

AUTOMATED ENTITLEMENT VERIFICATION FOR DELIVERY OF LICENSED SOFTWARE

5

PARTIAL WAIVER OF COPYRIGHT

CROSS REFERENCE TO RELATED APPLICATIONS
   Not Applicable

20   ## BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

   This invention generally relates to the field of software purchasing, distribution and help desk workflow, and more particularly to licensed software selected, delivered and installed through a network.

25   ### DESCRIPTION OF THE RELATED ART

   The deployment of software is the last step in the cycle of software commerce. It suffers from all of the classical product and product distribution problems. For most products the problems encountered include, manufacturing, documentation, shipping, storage, over production, inventory shortages, and theft. However, software has some

unique problems. One such problem is distribution. It's not uncommon for major software to be released more than once a year and distribution must be managed for frequent turnover where the product life is limited. In addition the distributors often are selling several competitive products. They are searching for a reason to sell one product over another product.

Once purchased the installation of the software product is often performed by consumers with limited computer skills. Many times this results in consumer confusion and frequent calls to help centers. The customer often times blames the software vendor for problems, which may be due to the particular PC being used, its configuration or the state of the co-resident software including the operating system. In a company the problems may be even worse as typically a IS (Information Services) person or department is accountable for not only all of the computers, which include hardware and software, but also the maintenance and upgrading. The employees always want the latest software and in many cases "require" special and unique configurations. Additionally the IS department tries to use just in time purchasing methods. Accordingly, a need exists for a efficient software deployment and installation solution to overcome the problems mentioned above.

Continuing further, there are two major developments that are influencing the software selection and deployment industry. One is the concept of very thin client, which usually means among other things, no removable media. With this type of a PC the software is installed and maintained and upgraded through a network. The second concept is the idea of "software is a service". Notable here is Microsoft's .NET strategy, and SUN Microsystems's .COM efforts. With these business models the concept is that software is not bought in the classical sense but rather it is rented, or even given away. Accordingly, a need exists for a distribution method for software products, to provide distribution for normal computers, for thin clients, normal software products and also support for software that is a service.

Medium to large corporations use a centrally managed, enterprise-wide, software delivery application to distribute software to employees in an automated manner using installation packages. The software is packaged in such a way that only certain approved options are offered. This allows for the installation to take place with minimal up-front

5 prompting of the user. The installation package is automated, and the installation performs identically for all employees. The installation package during installation will typically check for sufficient system resources on the employee's PC to stage, and perform the installation. Other checks are also performed including compatibility with the operating system, and other requisite software.

10

Although these centrally managed software delivery applications are useful they are not without their shortcomings. One shortcoming is the requirement to enter product keys during installation. Often times the management and security of these keys are problematic. Accordingly, a need exists for a method and a system to overcome this

15 problem.

Another shortcoming is the requirement for users to respond to installation prompts such as which installed options to select. Installation options such as destination directories and options are required. For many inexperienced computer users, the

20 answers to these installation questions are often not known. Moreover, different answers to installation questions among employees causes the installation package to install the software differently from system to system. Different installations across systems are much harder for support personnel to supply help and guidance. Accordingly, a need exists for a method and a system to overcome this problem.

25

Still, another shortcoming with the centrally managed, enterprise-wide, software delivery applications of the prior art is often times there is a need for internal or external logic associated with a software package to provide entitlement checks. The use of software entitlement checks with specialized programming and logic is expensive to

deploy and expensive to maintain. The administrative costs associated with entitlement are particularly expensive. Accordingly, a need exists to overcome this problem as well.

5   Yet, still another shortcoming with the centrally managed, enterprise-wide, software delivery applications is the amount of communications between licensed software applications and licensed management servers being installed and the server from which the software is downloaded. Many times even after software is downloaded in the prior art systems, license information must be passed back from the application to the server. Accordingly, a need exists to overcome this shortcoming as well.

10

SUMMARY OF THE INVENTION

Briefly, according to the present invention, disclosed is a method, a system and computer readable medium for the selection, configuration, entitlement, delivery, installation and maintenance of software over a network. The software is download with
15   associated scripting controls that will assure that all of the correct installation decisions and entitlement keys are entered. The software may be part of a major total upgrade bundle which includes several software product installations, or it may be a simple bug fix upgrade to existing software. This solution may be offered through a network for third party software vendors, or be part of a company wide software enterprise.

20

BRIEF DESCRIPTION OF THE DRAWINGS

The subject matter, which is regarded as the invention, is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the invention will be apparent from the
25   following detailed description taken in conjunction with the accompanying drawings.

FIG. 1 is a block diagram of an exemplary computer system that includes optional components, upon which the present invention can be implemented.

FIG. 2 is a block diagram of an exemplary software hierarchy that is executed on the hardware of FIG. 1.

FIG. 3 is a functional blockdiagram of a network connected client and server topology according to the present invention.

FIG. 4 is a block diagram of the payload that is delivered to the client as illustrated in FIG. 3 according to the present invention.

5   FIG. 5 is a flow diagram which shows the software delivery system performing the required checks, according to the present invention.

FIG. 6 is a flow diagram, which shows the over all ordering, delivery and installation of the requested software from a staging server, according to the present invention.

FIG. 7 is a screen shot of the selection of the desired software from a menu of

10   software according to the present invention.

FIG. 8 is a screen shot of the license agreement, which is presented to the client, according to the present invention.

FIG. 9 is a screen shot of a page which receives the entitlement ID which will enable the down load to proceed according to the present invention.

15   FIG. 10 is a screen shot of the installation progress screen, according to the present invention.

DETAILED DESCRIPTION OF AN EMBODIMENT

It is important to note, that these embodiments are only examples of the many

20   advantageous uses of the innovative teachings herein. In general, statements made in the specification of the present application do not necessarily limit any of the various claimed inventions. Moreover, some statements may apply to some inventive features but not to others. In general, unless otherwise indicated, singular elements may be in the plural and visa versa with no loss of generality.

25

In the drawing like numerals refer to like parts through several views.

EXEMPLARY SERVER SYSTEM

Referring to FIG. 1, there is shown a block diagram 100 of the major electronic components of an information processing system 100 for the one or more servers in accordance with the invention. The electronic components include: a central processing unit (CPU) 102, an Input/Output (I/O) Controller 104, a mouse 132 a keyboard 116, a system power and clock source 106; display driver 108; RAM 110, ROM 112, ASIC (application specific integrated circuit) 114 and a hard disk drive 118. These are representative components of a computer. The general operation of a computer comprising these elements is well understood. Network interface 120 provides connection to a computer network such as Ethernet over TCP/IP or other popular protocol network interfaces. Optional components for interfacing to external peripherals include: a Small Computer Systems Interface (SCSI) port 122 for attaching peripherals; a PCMCIA slot 124; and serial port 126. An optional diskette drive 128 is shown for loading or saving code to removable diskettes 130. The system 100 may be implemented by combination of hardware and software. Moreover, the functionality required for using the invention may be embodied in computer-readable media (such as 3.5 inch diskette 130) to be used in programming an information-processing apparatus (e.g., a personal computer) to perform in accordance with the invention.

EXAMPLE SOFTWARE HIERARCHY

FIG. 2 is a block diagram 200, illustrating the software hierarchy for the information processing system 100 of FIG. 1 according to the present invention. The BIOS (Basic Input Output System) 202 is a set of low level of computer hardware instructions for communications between an operating system 206, device driver 204 and hardware 200. Device drivers 204 are hardware specific code used to communicate between and operating system 206 and hardware peripherals such as a CD ROM drive or printer. Operating system 206 is the master program that loads after BIOS 202 initializes, that controls and runs the hardware 100. Examples of operating systems

include Windows 3.1/95/98/ME/2000/NT, Unix, Macintosh, OS/2, Sun Solaris and equivalents. Applications 208 are software application programs written in C/C++, assembler or other programming languages.

5   Applications 208 typically have several programs that are available for usage. Examples are software ordering system 210, a software delivery application 212, an entitlement ID database 214 and a deployment server application to transfer and install the ordered software 216. It is noted that there may be other programs in applications 208 such as word-processing files, spreadsheet files and network browsers (not 10 shown).

Taken together these are the preferred embodiments of the present invention. There are also other software files such as information 218. The information files can be such as, application 1 such as a word processing program 220, application 2 such as a 15 spread sheet program 222, and text information such as a application manual 224, or graphical information such as jpeg files 226.

DISCUSSION OF HARDWARE AND SOFTWARE IMPLEMENTATION OPTIONS

The present invention as would be known to one of ordinary skill in the art could be produced in hardware or software, or in a combination of hardware and software. 20 However in one embodiment the invention is implemented in software. The system, or method, according to the inventive principles as disclosed in connection with the preferred embodiment, may be produced in a single computer system having separate elements or means for performing the individual functions or steps described or claimed or one or more elements or means combining the performance of any of the functions or 25 steps disclosed or claimed, or may be arranged in a distributed computer system, interconnected by any suitable means as would be known by one of ordinary skill in art.

According to the inventive principles as disclosed in connection with the preferred embodiment, the invention and the inventive principles are not limited to any particular kind of computer system but may be used with any general purpose computer, as would be known to one of ordinary skill in the art, arranged to perform the functions described

5 and the method steps described. The operations of such a computer, as described above, may be according to a computer program contained on a medium for use in the operation or control of the computer, as would be known to one of ordinary skill in the art. The computer medium, which may be used to hold or contain the computer program product, may be a fixture of the computer such as an embedded memory or may be on

10 a transportable medium such as a disk, as would be known to one of ordinary skill in the art.

The invention is not limited to any particular computer program or logic or language, or instruction but may be practiced with any such suitable program, logic or

15 language, or instructions as would be known to one of ordinary skill in the art. Without limiting the principles of the disclosed invention any such computing system can include, inter alia, at least a computer readable medium allowing a computer to read data, instructions, messages or message packets, and other computer readable information from the computer readable medium. The computer readable medium may

20 include non-volatile memory, such as ROM, Flash memory, floppy disk, Disk drive memory, CD-ROM, and other permanent storage. Additionally, a computer readable medium may include, for example, volatile storage such as RAM, buffers, cache memory, and network circuits.

25 Furthermore, the computer readable medium may include computer readable information in a transitory state medium such as a network link and/or a network interface, including a wired network or a wireless network, that allow a computer to read such computer readable information.

SOFTWARE ENTITLEMENT

FIG. 3 illustrates a functional flow block diagram 300, of a network 302 with attached clients and servers. Client 1 304 through client N 308 are connected to a network 302 such as the Internet. The ordering process over a network is described

5   below.

This process begins with an employee or customer using Client 1 304 orders 308 a copy of a particular software package. The order is received at the software ordering system 310. This ordering system contains a list of available software from the

10   employee's company and / or any third party software. Once the order is placed the ID is sent 312 back to Client 1 304. This same entitlement ID # is also sent 316 to the software delivery application 314. The entitlement ID # is stored in the software delivery application's entitlement ID database 322. Once Client 1 304 is ready to receive the ordered software, a request is made 318, to the software delivery application 314. This

15   request may be made as soon as the entitlement ID # is received by the Client 1 304, or at a time later. This request is received by the software delivery application server 320, which is part of the software delivery application 314. The software delivery application 320 compares 324 the received entitlement ID # to the one that is stored in the entitlement ID database 322. Assuming that the ID #s match the delivery server 320

20   schedules 328 the ordered software on the staging servers 326. If the ID #s do not match, an error message is sent back to the Client 1 (not shown). Finally the correct staging server 326 connects with the client 1 304 and the ordered software payload is downloaded and installed 330.

25   SOFTWARE PAYLOAD DESCRIPTION

Turning now to FIG. 4, illustrated is a block diagram 400 of the requested software that is delivered, as a payload 402 using a network 302, to the client 1 304 by the staging server 326. The overall script 404 contains all of the information required to download and

install the requested software. The software may consist of only one file or as shown software files 1 418 through N 422. It is noted that the software may be any type of file. It may be data as in text files, spreadsheets, databases, presentations, or multimedia files. The software may be also be executable files such as word processors, spreadsheet

5 programs, database programs or presentation software. Moreover, the software may be a collection of software bundled together as a standard business software "seat" or a workstation "bench" in the case of science and engineering software.

The over all script 404 in this example of the present invention is controlling

10 individual installation scripts 1 406 through script N 410. These scripts 1 through N install the requested software 418 through 422 into Client 1 304 if and only if the software keys 1 through N412 through 416 are verified. In some cases a key may not be required for download and installation. These software keys contain software security codes, which the scripting controls 406 through 410 will control during the installation. The ordered software

15 is downloaded as pre-selected images or "disks" 418 through 422 that are known to be compatible with the client's hardware and software, and that were verified during the installation process. The client's owner or operator need not be involved with any phase of the down load or installation. No further selections or interactions are required. No questions are asked, such as: "Where should the file be stored?", "select options" or

20 selections that involve any of the possibly confusing decisions or opportunities for mistakes. All of the registration fill-in information has already been performed. Once initiated, the payload as described here causes the software to be downloaded and installed without any operator intervention.

FURTHER DISCUSSION OF PAYLOAD

25 It is important to note that neither the software product being installed "disk" 1 through N 418 through 422 nor the software scripting such as the overall script 404 or the individual scripts 1 through N 406 - 410 require any internal or external logic associated with them to provide the entitlement check. The entitlement is performed by

the software delivery application independently of the software package without impact to the software package repository.

5   Administrative costs associated with entitlement at the software delivery application are greatly minimized. The administrator is only required to define a relationship between the software package, that is, an identifier within the software delivery application 314 for the package and the software product "disk" 1 through N 418 - 422. This enables any corresponding entitlement IDs received from the software delivery application 314 to be associated with the payload package 402 in an automated

10   manner.

     Finally understand that the startup and shutdown, behavior, and use of the software product "disk" 1 through N 418 - 422 once installed, is not affected in any way. No communication is required between the software product and a license management

15 server at the time of execution. No client entitlement, or license management code or API is associated with the software product.

ENTITLEMENT VERIFICATION OF CLIENT'S SOFTWARE SELECTION

     Turning now to FIG. 5, illustrated is a flow diagram 500, which illustrates the

20 verification process. The flow is entered 502 when a client, such as 304, decides to order a copy of a software file. The client makes a request 504 for software from a "standard" set of offerings. The software vendor ordering system 310 checks 506 to see if the client is a standard PC platform. This standardization is required in a company to assure a level of similarity between client PCs of a given type. Given the standardization

25 of the software, checks are performed to assure compatibility with the ordered software. A standard installation on the company's PCs reduces help desk maintenance. If the client PC platform is not within the predefined profile the flow is exited 516 and the software is not delivered. If the PC platform standard compliance is verified 506, the

software vendor ordering system 310 checks if sufficient storage space is available 508. If the ordered software is too large for the client's PC a download will not be scheduled. If there is insufficient space the flow is exited 516. If there is space then the software is checked 510. The version of the operating system and any co-resident software that is

5 required is checked. Here again the software that is being ordered will not work if the operating system is not of the right version and release, and similarly for the support software 512. If the software check is failed the flow is exited 516. If the check for co-resident software is passed then the ordering system 310 can install the requested software 514 and the flow is exited 516.

10 <u>SOFTWARE ENTITLEMENT FLOW DIAGRAM.</u>

Turning now to FIG. 6, flow diagram 600 is shown which illustrates the ordering and the fulfillment of the present invention. The flow is entered 602 when a client orders software 500 from the software ordering system 310. It is further assumed that the entitlement ID has been sent to the client 606. In addition the software vendor ordering

15 system 310 sends notification to the software delivery application 314 with the entitlement ID # 608. This entitlement ID # is saved into a database 322 at the software delivery application 610. At any time after receiving the entitlement ID # the client can access the software delivery application 612 and request the delivery of the ordered software 614. The client is presented with a license agreement 616. If the agreement is

20 not accepted the flow is exited 626. If the agreement is accepted 616 then the client can enter the entitlement ID # 618 If the entered entitlement ID # does not match the stored entitlement ID # in the entitlement ID database 322 the flow is exited 626. If it does match 620 then the software delivery application schedules a delivery 622 via the staging servers 326. The software delivery application 314 next performs the pre-

25 installation audits 623, which are explained below. Finally the software delivery application downloads and installs 624 the requested software and the flow is exited 626.

SCREEN SHOT OF SOFTWARE SELECTION.

Turning now to FIG.7, is a screen shot 700 of the present invention's software selection page. Using a Internet browser 702 the client is presented the page entitled IBM standard software - W/2000 for North America. Note that a geographic selection

5   was made earlier. (not shown) This Internet page contains the normal Windows controls 704 such as file, edit, view, favorites, tools, and help. The client has selected a URL (Uniform Resource Locator) that presents the W3 IBM standard software installer selection menu 706. From this selection menu the client has selected the W/2000 page from the W3 home page 708. This is indicated, as "W/2000" is bold. Given this selection

10   the client is presented with a search catalog and a list of topics, from which the client has selected the advanced technology applications 710. This is indicated, as "as-is applications" is bold. This menu selection in turn has presented the client with the main menu of as-is applications 712. From this main menu the client has selected the Microsoft Project 2000 software package for download and installation. Finally the lower

15   task bar 714 indicates that the client is connected to the Internet and that the transfer is completed.

SCREEN SHOT OF ENTITLEMENT ID ENTRY PAGE

FIG. 8, is a screen shot 800 of the start of the licensing agreement, according to the present invention. Once the client has selected the desired software a dialog is

20   presented to assure that the client agrees to the licensing and has local permission for the desired software. A typical browser window contains the Internet page 802 is shown, which is presented when the client selects a desired program Microsoft Project 2000 for download and installation 712. This Internet page contains the typical Windows controls 804 such as file, edit, view, favorites, tools, and help. The client has selected a

25   URL (Uniform Resource Locator) that presents the W3 IBM standard software installer selection menu 706, and is continuing to select additional sub-menus. The client has selected the W/2000 page from the W3 home page 808. On the previous screen the client selected desired software. This software, the Microsoft Project 2000 is presented

with a legal notice and licensing agreement 812. The client reads the agreement and accepts it 616, which requires scrolling down, (not shown) and at the end of this agreement certain information will be requested. The information is the entitlement ID #. Finally the lower task bar 814 indicates that the client is connected to the Internet and

5   that the transfer of the present page is completed.

SCREEN SHOT OF LICENSING AGREEMENT PURCHASE AND ORDER NUMBERS

Turning now to FIG. 9, which shows a screen shot 900 of the entitlement ID entry page , according to the present invention. A normal Browser window presents Internet

10   page 902. This Internet page contains the normal Windows controls 904 such as file, edit, view, favorites, tools, and help. The client has selected a URL (Uniform Resource Locator) that presents the W3 IBM standard software installer selection menu 906, and is continuing to select additional sub-menus. The client has selected the W/2000 page from the W3 home page 908. On the previous screen the client was presented with the

15   licensing agreement shown as screen shot 800, which the client has read and agreed to 616. This page presents the client with the details of the selected software 910. The required operating system is listed 912. The disk space required to download and install the selected software is listed 914, which includes the temporary requirement and then the finally amount that is needed.

20   Next the entitlement ID 916 must be entered. Optionally the location of temporary drive space may be entered 918. Assuming that the entitlement ID has been entered and that it is found in the software delivery application's database 314, the download of the selected software is scheduled by the delivery application server 320, at the staging

25   servers 326 once the "Install Now !" button is selected 922.

Finally the lower task bar 922 indicates that the client is connected to the Internet and that the transfer of the present page is completed.

POU920010051US1   14 of 24

<u>SCREEN SHOT OF THE SOFTWARE INSTALLER</u>

Turning now to FIG. 10, which shows a screen shot 1000 of the installation of the requested software, according to the present invention. Assuming that the entitlement ID has been checked successfully during screen shot 900, the installation commences. The scheduling server 326 established a download session during which the requested software payload 402 with the associated scripts and entitlement ID # will be downloaded. A dialog box 1002 is presented to indicate the steps and the progress of the installation. This Window contains the normal Windows controls 1004 such as file, windows and help. In this view the staging servers have started the installation 1006 once the install code has been assembled into the payload, and as show, this operation is 29% compete. Once this is complete several checks are performed by software delivery application: verify the PC Platform 506, the storage space 508, the Operating System 510 and the prerequisite software 512. 1008. Once the product is downloaded and installed the temporary files are removed 1010. It is noted that no configuration questions are asked or options are provided. Once started the present invention completes the download and installation in an efficient and unattended way, which obviates incorrect or even different choices by the client. Any future help desk support can be accomplished with a known level of consistency. Finally during the installation the client may cancel the installation operation using the cancel button 1012.

<u>NON-LIMITING EXAMPLES</u>

Although a specific embodiment of the invention has been disclosed. It will be understood by those having skill in the art that changes can be made to this specific embodiment without departing from the spirit and scope of the invention. The scope of the invention is not to be restricted, therefore, to the specific embodiment, and it is intended that the appended claims cover any and all such applications, modifications, and embodiments within the scope of the present invention.

What is claimed is: